

Práctica 1

Configuración de un router para la provisión de Calidad de servicio según SLA

0. Primero vamos a configurar el entorno de experimentación como hicimos en el seminario 1.

Host1

```
root@manureina-virtualbox:/home/manureina# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::8b93:9dfe:2361:f14 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c7:3e:b2 txqueuelen 1000 (Ethernet)
    RX packets 162490 bytes 220095219 (220.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 49531 bytes 3012424 (3.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.7.1.51 netmask 255.255.255.0 broadcast 10.7.1.255
    ether 08:00:27:0d:91:b4 txqueuelen 1000 (Ethernet)
    RX packets 541 bytes 58135 (58.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14988 bytes 1733502 (1.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 1298 bytes 107886 (107.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1298 bytes 107886 (107.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@manureina-virtualbox:/home/manureina# route -n
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref      Uso Interfaz
0.0.0.0      10.7.1.110    0.0.0.0      UG    0      0      0 enp0s8
0.0.0.0      10.0.2.2      0.0.0.0      UG    100    0      0 enp0s3
10.0.2.0     0.0.0.0       255.255.255.0 U      100    0      0 enp0s3
10.7.1.0     0.0.0.0       255.255.255.0 U      0      0      0 enp0s8
```

Host2

```
root@manureina-virtualbox:/home/manureina# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::3765:1f05:e87b:326 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:45:46:a1 txqueuelen 1000 (Ethernet)
    RX packets 1451 bytes 1310690 (1.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1176 bytes 96005 (96.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.1.21 netmask 255.255.255.0 broadcast 172.17.1.255
    ether 08:00:27:c2:d6:f6 txqueuelen 1000 (Ethernet)
    RX packets 4180 bytes 435333 (435.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1045 bytes 144090 (144.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 1276 bytes 105592 (105.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1276 bytes 105592 (105.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@manureina-virtualbox:/home/manureina# route -n
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref      Uso Interfaz
0.0.0.0      172.17.1.110  0.0.0.0      UG     0      0      0 enp0s8
0.0.0.0      10.0.2.2      0.0.0.0      UG     100     0      0 enp0s3
10.0.2.0     0.0.0.0       255.255.255.0 U      100     0      0 enp0s3
172.17.1.0   0.0.0.0       255.255.255.0 U      0      0      0 enp0s8
```

Router

```
vyos@vyos# show int
 ethernet eth0 {
     address 10.7.1.110/24
     hw-id 08:00:27:08:6a:b4
 }
 ethernet eth1 {
     address 172.17.1.110/24
     hw-id 08:00:27:8a:8a:27
 }
 loopback lo {
 }
[edit]
vyos@vyos#
```

```

root@manureina-virtualbox:/home/manureina# route -n
Tabla de rutas IP del núcleo

```

Destino	Pasarela	Genmask	Indic	Métric	Ref	Uso	Interfaz
0.0.0.0	10.7.1.110	0.0.0.0	UG	0	0	0	enp0s8
0.0.0.0	10.0.2.2	0.0.0.0	UG	100	0	0	enp0s3
10.0.2.0	0.0.0.0	255.255.255.0	U	100	0	0	enp0s3
10.7.1.0	0.0.0.0	255.255.255.0	U	0	0	0	enp0s8

```

root@manureina-virtualbox:/home/manureina#

```

Pasarela del host1 con la eth0 del router

```

root@manureina-virtualbox:/home/manureina# route -n
Tabla de rutas IP del núcleo

```

Destino	Pasarela	Genmask	Indic	Métric	Ref	Uso	Interfaz
0.0.0.0	172.17.1.110	0.0.0.0	UG	0	0	0	enp0s8
0.0.0.0	10.0.2.2	0.0.0.0	UG	100	0	0	enp0s3
10.0.2.0	0.0.0.0	255.255.255.0	U	100	0	0	enp0s3
172.17.1.0	0.0.0.0	255.255.255.0	U	0	0	0	enp0s8

```

root@manureina-virtualbox:/home/manureina#

```

Pasarela del host2 con la eth1 del router.

Configurados las IPs del router, las IP y gateways de los hosts solo queda comprobar el funcionamiento desde un host a otro con un ping o un traceroute.

Ping desde el router a ambos host.

```

vyos@vyos# ping 10.7.1.51
PING 10.7.1.51 (10.7.1.51) 56(84) bytes of data.
64 bytes from 10.7.1.51: icmp_seq=1 ttl=64 time=0.741 ms
64 bytes from 10.7.1.51: icmp_seq=2 ttl=64 time=0.867 ms
64 bytes from 10.7.1.51: icmp_seq=3 ttl=64 time=0.830 ms
^C
--- 10.7.1.51 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 0.741/0.812/0.867/0.062 ms
[edit]
vyos@vyos# ping 172.17.1.21
PING 172.17.1.21 (172.17.1.21) 56(84) bytes of data.
64 bytes from 172.17.1.21: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 172.17.1.21: icmp_seq=2 ttl=64 time=0.888 ms
64 bytes from 172.17.1.21: icmp_seq=3 ttl=64 time=0.971 ms
^C
--- 172.17.1.21 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 0.888/0.960/1.023/0.066 ms
[edit]
vyos@vyos#

```

Traceroute desde el host1 al host2 (vemos como realiza la ruta en dos pasos, primero se dirige al router mediante la pasarela host1-router y del router al host2).

```
root@manureina-virtualbox:/home/manureina# traceroute 172.17.1.21
traceroute to 172.17.1.21 (172.17.1.21), 64 hops max
 1  10.7.1.110  1,037ms  0,509ms  0,632ms
 2  172.17.1.21  0,841ms  0,854ms  0,663ms
root@manureina-virtualbox:/home/manureina#
```

1. Caracterice el tráfico que generan dos aplicaciones, mediante el uso del generador de tráfico ITG y el analizador de tráfico wireshark. Para ello, siga las especificaciones indicadas en el apartado: “A.1. Caracterización de distintos tipos de tráfico”. La sintaxis básica del generador de tráfico puede consultarla en la sección “B. Sintaxis del generador de tráfico D-ITG”. Debe contestar a las preguntas 1 :

He creado un script.

```
-a 172.17.1.21 -rp 3000 -t 20000 VoIP
-a 172.17.1.21 -rp 3000 -t 20000 Quake3
```

```
root@manureina-virtualbox:/home/manureina# ITGSend script.txt -l script.log
ITGSend version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Voice Codec: G.711.1
Framesize: 80.00
Samples: 1
Packets per sec.: 100
VAD: No
Started sending packets of flow ID: 2
Started sending packets of flow ID: 1
Finished sending packets of flow ID: 1
Finished sending packets of flow ID: 2
root@manureina-virtualbox:/home/manureina#
```

```

root@manureina-virtualbox:/home/manureina# ITGRecv
ITGRecv version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Press Ctrl-C to terminate
Listening on UDP port : 3000
Finish on UDP port : 3000
Finish on UDP port : 3000

```

a) ¿Que tasa de transferencia media y máxima requiere la aplicación Quake3?

Generamos el tráfico de la aplicación Quake3.

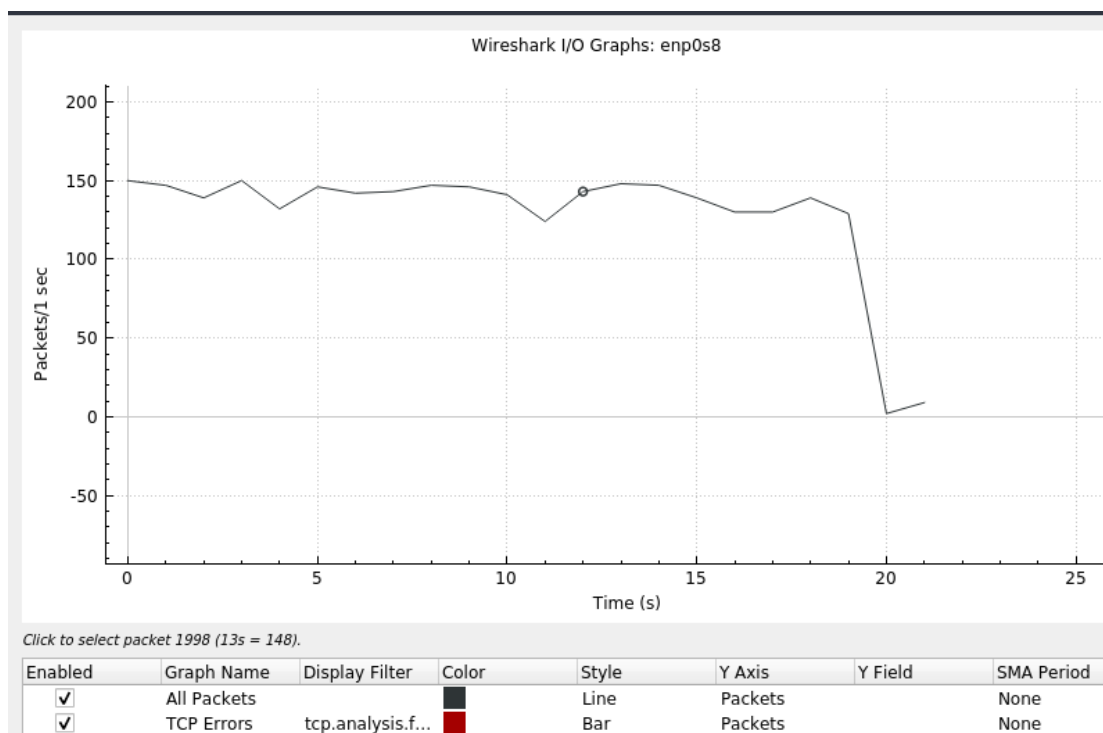
```

root@manureina-virtualbox:/home/manureina# ITGSend -a 172.17.1.21 -rp 3001 -t 20000 -l scrip1.log Quake3
ITGSend version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Started sending packets of flow ID: 1
Finished sending packets of flow ID: 1

root@manureina-virtualbox:/home/manureina#

```

Como vemos en la gráfica de tráfico de Quake 3, tiene una tasa máxima de unos 150 packets/second y una tasa media de 140 packets/second aprox.



Reina Alonso

b) ¿Que tasa de transferencia media y máxima requiere la aplicación de VoIP?

Generamos tráfico de la aplicación VoIP.

```
root@manureina-virtualbox:/home/manureina# ITGSend -a 172.17.1.21 -rp 3000 -t 20000 -l scrip1.log VoIP
```

Vemos una tasa de transferencia máxima de 250 packets/second y una tasa media de transferencia de 175 packets/second aprox.



2. Ejecute el generador de paquetes ITG para simular una fuente de tráfico elástico y otra con tráfico inelástico. Configure los routers según se especifica en el apartado: “A.2. Tráfico elástico e inelástico”. La sintaxis básica del generador de tráfico puede consultarla en la sección “B. Sintaxis del generador de tráfico D-ITG”. Debe contestar a la siguiente pregunta:

Primero vamos a limitar la tasa que permite el enlace entre el router y el segundo ordenador a 2Mbps. De esta forma se simula un enlace real con una tasa máxima. Para ello se puede utilizar una política de limitación en la interfaz del router dentro de la configuración del mismo:

```
vyos@vyos# set interfaces ethernet eth1 traffic-policy out limitadorenlace
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot' ...
Done
[edit]
vyos@vyos# set traffic-policy shaper limitadorenlace default bandwidth 2mbps
```

```
-a 172.17.1.21 -rp 3000 -t 20000 -T UDP -C 300 VoIP
-a 172.17.1.21 -rp 3000 -t 20000 -T TCP -C 300 Quake3
```

```
root@manureina-virtualbox:/home/manureina# ITGSend script.txt -l script.log
ITGSend version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Voice Codec: G.711.1
Framesize: 80.00
Samples: 1
Packets per sec.: 100
VAD: No
Started sending packets of flow ID: 2
Started sending packets of flow ID: 1
Finished sending packets of flow ID: 2
Finished sending packets of flow ID: 1
root@manureina-virtualbox:/home/manureina#
```

Time	Source	Destination	Protocol	Length	Info
1 0.000000000	10.7.1.51	172.17.1.21	TCP	74	43342 → 9000 [SYN] Seq=0 Win=64240
2 0.000007403	172.17.1.21	10.7.1.51	TCP	74	9000 → 43342 [SYN, ACK] Seq=0 Ack=1
3 0.001880995	10.7.1.51	172.17.1.21	TCP	66	43342 → 9000 [ACK] Seq=1 Ack=1 Win=6
4 0.002294160	10.7.1.51	172.17.1.21	TCP	67	43342 → 9000 [PSH, ACK] Seq=1 Ack=1
5 0.002321497	172.17.1.21	10.7.1.51	TCP	66	9000 → 43342 [ACK] Seq=1 Ack=2 Win=6
6 0.015624122	172.17.1.21	10.7.1.51	TCP	67	9000 → 43342 [PSH, ACK] Seq=1 Ack=2
7 0.016844953	10.7.1.51	172.17.1.21	TCP	66	43342 → 9000 [ACK] Seq=2 Ack=2 Win=6
8 0.017306091	10.7.1.51	172.17.1.21	S101	93	43342 → 9000 [PSH, ACK] Seq=2 Ack=2
9 0.017342281	172.17.1.21	10.7.1.51	TCP	66	9000 → 43342 [ACK] Seq=2 Ack=29 Win=6
10 0.018280298	10.7.1.51	172.17.1.21	S101	93	43342 → 9000 [PSH, ACK] Seq=29 Ack=2
11 0.018309234	172.17.1.21	10.7.1.51	TCP	66	9000 → 43342 [ACK] Seq=2 Ack=56 Win=6
12 0.019330017	172.17.1.21	10.7.1.51	S101	73	9000 → 43342 [PSH, ACK] Seq=2 Ack=56
13 0.020366251	10.7.1.51	172.17.1.21	TCP	66	43342 → 9000 [ACK] Seq=56 Ack=9 Win=6

a) ¿Qué ocurre cuando ambos tipos de tráfico pasan por en el mismo router cuando se produce congestión?

Se produce una pérdida de paquetes ya que no puede atender ambas peticiones.

3. - Genere varios flujos TCP que compitan por el ancho de banda disponible.

Generamos el script con varios flujos.

```
-a 172.17.1.21 -rp 3000 -t 20000 -C 3000 -T TCP Quake3
-a 172.17.1.21 -rp 3001 -t 20000 -C 4000 -T TCP Quake3
-a 172.17.1.21 -rp 3002 -t 20000 -C 5000 -T TCP Quake3
```

```
root@manureina-virtualbox:/home/manureina# vi script2.txt
root@manureina-virtualbox:/home/manureina# ITGSend script2.txt -l script2.log
ITGSend version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Started sending packets of flow ID: 3
Started sending packets of flow ID: 1
Started sending packets of flow ID: 2
```


Time	Source	Destination	Protocol	Length	Info
8315	21.048229533	10.7.1.51	172.17.1.21	TCP	66 43360 → 9000 [ACK] Seq=98 Ack=28
8316	22.046281412	172.17.1.21	10.7.1.51	TCP	66 43360 → 43360 [PSH, ACK] Seq=28 Ack=38
8317	22.047952603	10.7.1.51	172.17.1.21	TCP	66 43360 → 9000 [ACK] Seq=98 Ack=38
8318	23.046807898	172.17.1.21	10.7.1.51	TCP	66 43360 → 43360 [PSH, ACK] Seq=33 Ack=38
8319	23.050148666	10.7.1.51	172.17.1.21	TCP	66 43360 → 9000 [ACK] Seq=98 Ack=38
8320	23.050149665	10.7.1.51	172.17.1.21	TCP	67 43360 → 9000 [PSH, ACK] Seq=98 Ack=38
8321	23.050217755	172.17.1.21	10.7.1.51	TCP	66 9000 → 43360 [ACK] Seq=38 Ack=98
8322	23.050570160	172.17.1.21	10.7.1.51	TCP	67 9000 → 43360 [PSH, ACK] Seq=38 Ack=98
8323	23.050649051	172.17.1.21	10.7.1.51	TCP	66 9000 → 43360 [FIN, ACK] Seq=39 Ack=98
8324	23.052318431	10.7.1.51	172.17.1.21	TCP	66 43360 → 9000 [ACK] Seq=99 Ack=38
8325	23.053585473	10.7.1.51	172.17.1.21	TCP	66 43360 → 9000 [FIN, ACK] Seq=99 Ack=38
8326	23.053630703	172.17.1.21	10.7.1.51	TCP	66 9000 → 43360 [ACK] Seq=40 Ack=100

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s8, id 0
 ▶ Ethernet II, Src: PcsCompu_8a:8a:27 (08:00:27:8a:8a:27), Dst: PcsCompu_c2:d6:f6 (08:00:27:c2:d6:f6)
 ▶ Internet Protocol Version 4, Src: 10.7.1.51, Dst: 172.17.1.21
 ▶ Transmission Control Protocol, Src Port: 43360, Dst Port: 9000, Seq: 0, Len: 0

a) ¿Qué ocurre? ¿Cuál obtiene mayor tasa de transmisión?

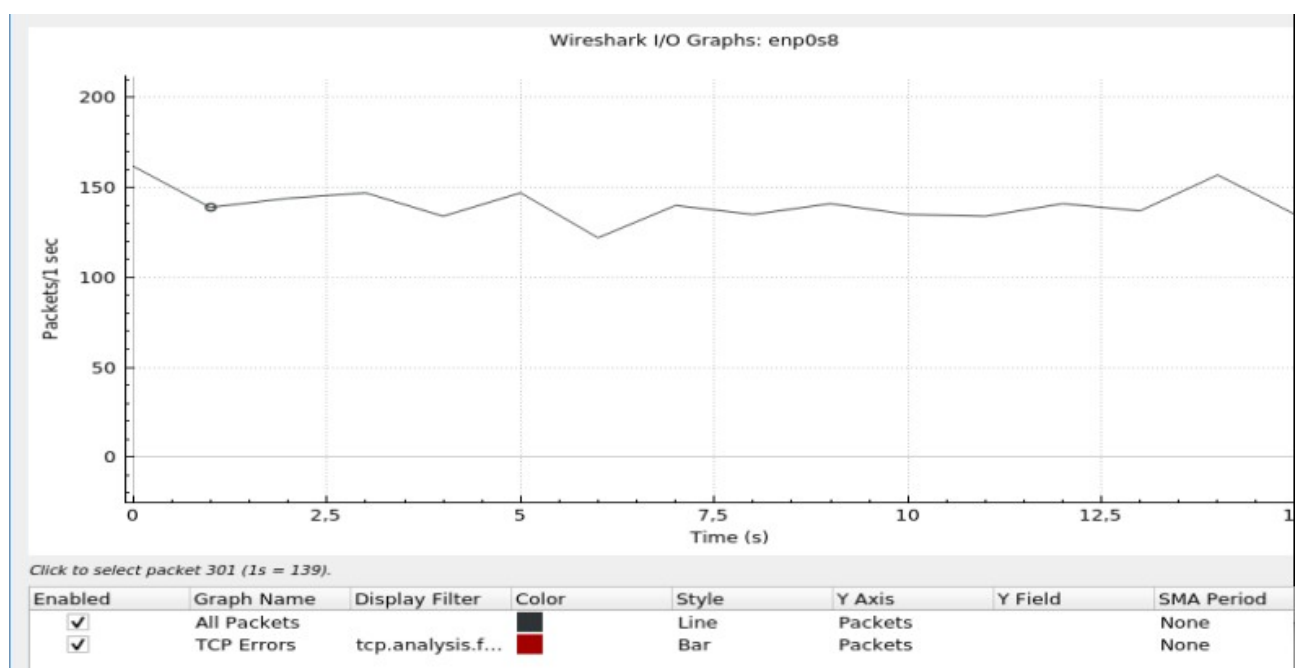
***** TOTAL RESULTS *****		
Number of flows	=	3
Total time	=	19.999673 s
Total packets	=	8259
Minimum delay	=	0.000000 s
Maximum delay	=	0.000000 s
Average delay	=	0.000000 s
Average jitter	=	0.000000 s
Delay standard deviation	=	0.000000 s
Bytes received	=	525464
Average bitrate	=	210.189037 Kbit/s
Average packet rate	=	412.956752 pkt/s
Packets dropped	=	0 (0.00 %)
Average loss-burst size	=	0 pkt
Error lines	=	0

Todos tienen una tasa de transmisión parecida pero el que mayor tiene por poco es el flow number: 2.

4. Compare el efecto de configurar en el router una política de verificación de conformidad (policing) y un esquema de conformado de tráfico (shaping) para un mismo flujo de tráfico. Configure los routers según se especifica en el apartado: “A.3. Efecto de policing y shaping”. (Vea además las secciones “C. Configuración de clases de tráfico”, “C.1. Configuración de policing” y “C.2. Configuración de shaper puro”). Debe contestar a la siguiente cuestión:

1.- Genere con ITGSend un flujo de tráfico Quake3, con puerto de destino 6666.

```
ta 172.17.1.21 -rp 6666 -t 20000 Quake3
```



2.- Defina la clase de tráfico QUAKE3, con el tráfico UDP que vaya dirigido al puerto 6666.

```
vyos@vyos# set traffic-policy shaper ejercicio4 class 30 match QUAKE3 ip port 6666_
```

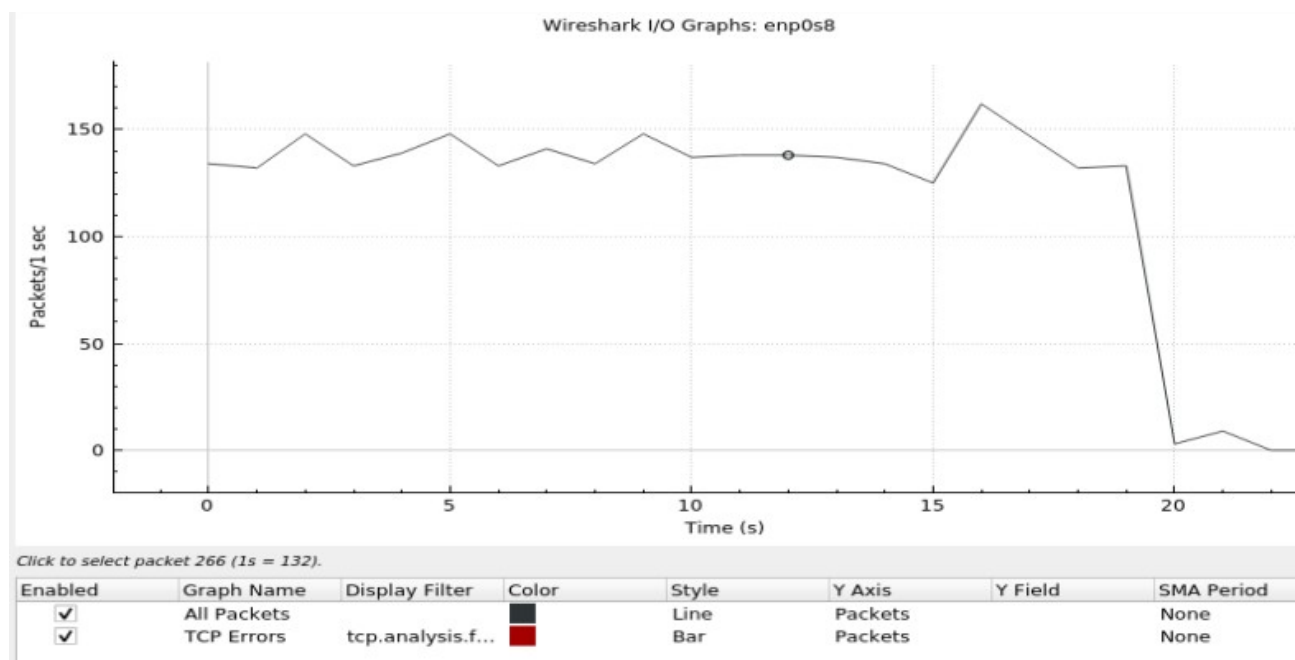
```
vyos@vyos# set traffic-policy shaper ejercicio4 class 30 match QUAKE3 ip protocol udp
```

```
vyos@vyos# sh traffic
+shaper ejercicio4 {
+  class 30 {
+    match QUAKE3 {
+      ip {
+        protocol udp
+        source {
+          port 6666
+        }
+      }
+    }
+  }
+}
+}
```

3.- Asigne una política de QoS con police a 112000 bps y asígnesela a la interfaz Eth0 (limiter) o interfaz Eth1 (shaper) del router. (Consultar “C.1. Configuración de policing”)

```
vyos@vyos# set traffic-policy shaper ejercicio4 default bandwidth 112000bps
[edit]
```

```
vyos@vyos# set interfaces ethernet eth1 traffic-policy out ejercicio4_
```

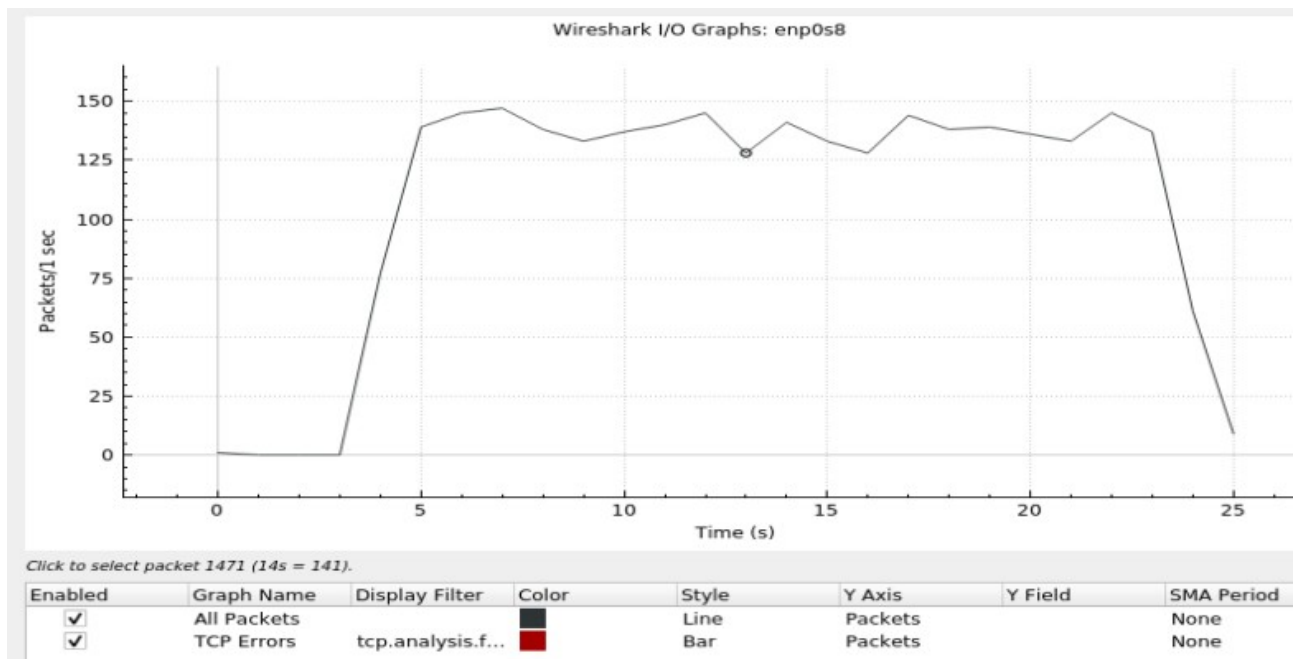


4.- Asigne una política de QoS con shape a 112000 bps y asígnesela a la interfaz Eth1 del router.

```
vyos@vyos# set traffic-policy rate-control myshaper bandwidth 112000bps_
```

```
rate-control myshaper {  
    bandwidth 112000bps  
}
```

```
[edit]  
vyos@vyos# sh int  
ethernet eth0 {  
    address 10.7.1.110/24  
    hw-id 08:00:27:08:6a:b4  
}  
ethernet eth1 {  
    address 172.17.1.110/24  
    hw-id 08:00:27:8a:8a:27  
    traffic-policy {  
        out myshaper  
    }  
}  
loopback lo {  
}  
[edit]  
vyos@vyos# _
```



No.	Time	Source	Destination	Protocol	Length	Info
16	0.044324559	10.7.1.51	172.17.1.21	UDP	105	48366 → 6666 Len=63
17	0.048722374	10.7.1.51	172.17.1.21	UDP	106	48366 → 6666 Len=64
18	0.059910260	10.7.1.51	172.17.1.21	UDP	99	48366 → 6666 Len=57
19	0.070856601	10.7.1.51	172.17.1.21	UDP	113	48366 → 6666 Len=71
20	0.071970195	10.7.1.51	172.17.1.21	UDP	103	48366 → 6666 Len=61
21	0.083148249	10.7.1.51	172.17.1.21	UDP	108	48366 → 6666 Len=66
22	0.094988378	10.7.1.51	172.17.1.21	UDP	104	48366 → 6666 Len=62
23	0.103080335	10.7.1.51	172.17.1.21	UDP	106	48366 → 6666 Len=64

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s8, id 0
 ▶ Ethernet II, Src: PcsCompu_8a:8a:27 (08:00:27:8a:8a:27), Dst: PcsCompu_c2:d6:f6 (08:00:27:c2:d6:f6)
 ▶ Internet Protocol Version 4, Src: 10.7.1.51, Dst: 172.17.1.21
 ▶ Transmission Control Protocol, Src Port: 43348, Dst Port: 9000, Seq: 0, Len: 0

0000	08 00 27 c2 d6 f6	08 00 27 8a 8a 27	08 00 45 00E..
0010	00 3c 93 a8 40 00	3f 06 ef b3 0a 07	01 33 ac 11	<...@?....3..
0020	01 15 a9 54 23 28	46 96 8f a5 00 00	00 00 a0 02	...T#(F.....
0030	fa f0 82 c6 00 00	02 04 05 b4 04 02	08 0a 02 ce
0040	6c 62 00 00 00 00	01 03 03 07		1b.....

```

Flow number: 1
From 10.7.1.51:43623
To 172.17.1.21:6666
-----
Total time = 19.998391 s
Total packets = 2716
Minimum delay = 0.000000 s
Maximum delay = 0.000000 s
Average delay = 0.000000 s
Average jitter = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received = 172948
Average bitrate = 69.184766 Kbit/s
Average packet rate = 135.810926 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0.000000 pkt
-----

***** TOTAL RESULTS *****
-----
Number of flows = 1
Total time = 19.998391 s
Total packets = 2716
Minimum delay = 0.000000 s
Maximum delay = 0.000000 s
Average delay = 0.000000 s
Average jitter = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received = 172948
Average bitrate = 69.184766 Kbit/s
Average packet rate = 135.810926 pkt/s
Packets dropped = 0 (0.00 %)

```

a) ¿Qué diferencia observa en el tráfico tras usar policing y shaping?

Lo primero que podemos observar es que antes de hacer policing y shaping, la tasa de packets/s era mayor (superaba en algunos puntos una tasa de 150), mientras que tras aplicar policing y shaping la tasa queda limitada. En el caso de policing únicamente en un pico consigue superar esa tasa, por otro lado con shaping nunca llega a esa tasa, siempre queda por debajo.

Por otro lado con policing vemos que se producen más picos que sin ninguna política y también más picos respecto de shaping. Esto se debe a que, con policing, comprobamos si el tráfico de paquetes cumple unas especificaciones de tráfico y si supera los límites se lleva acabo una acción sobre el paquete recibido y así se puede controlar el flujo de tráfico que llega a una red.

Mientras que con shaping podemos modificar el tráfico de salida de un router para que los paquetes sigan un tráfico regular, sin picos. En la imagen de shaping, podemos ver como al principio, tiene una tasa muy baja, luego se produce una subida muy rápida y se mantiene una tasa regular durante la mayor parte del tiempo, para posteriormente, volver a bajar la tasa muy rápidamente.

5. Configure el router para que ofrezca la QoS requerida por el acuerdo de nivel de servicio (SLA, Service Level Agreement) indicado en el apartado: "A.4. Configuración de SLA básico". Compruebe dicha configuración. Debe especificar razonadamente la configuración que propone y evaluarla.

Un SLA define el contrato entre cliente y proveedor de servicios. En esta práctica vamos a configurar el router para que aplique un SLA que satisfaga las siguientes condiciones:

- El tráfico de voz sobre IP (VoIP) tendrá prioridad, con 80 kbps garantizados.
- El tráfico de Quake3 estará conformado a 112 kbps.
- El resto del tráfico se debe limitar a 2 Mbps. Cualquier tráfico que sobre pase estos umbrales debe ser descartado.

```
shaper ejercicio5 {  
  class 20 {  
    bandwidth 80kbps  
    match voip {  
      ip {  
        destination {  
          port 6667  
        }  
      }  
    }  
    priority 3  
  }  
  class 30 {  
    bandwidth 112kbps  
    match quake3 {  
      ip {  
        destination {  
          port 6666  
        }  
      }  
    }  
    priority 4  
  }  
  default {
```

```
}  
default {  
  bandwidth 2mbps  
  ceiling 100%  
}
```

```

vyos@vyos# set interfaces ethernet eth1 traffic-policy out ejercicio5
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@vyos# sh int
  ethernet eth0 {
    address 10.7.1.110/24
    hw-id 08:00:27:08:6a:b4
  }
  ethernet eth1 {
    address 172.17.1.110/24
    hw-id 08:00:27:8a:8a:27
    traffic-policy {
      out ejercicio5
    }
  }
  loopback lo {
  }
[edit]

```

Para la configuración he creado un shaper llamado ejercicio5 y le he asignado un bandwidth por default de 2mpbs y un ceiling del 100%. Posteriormente he creado dos clases, una para identificar el tráfico de VoIP y otra para el de Quake3. A la clase de VoIP le he asignado una mayor prioridad que a la clase de quake3 y un bandwidth de 80kbps, va a identificar este tráfico porque será mandado por el puerto 6667.

Para la clase que identifica quake3 le he asignado un ID 30, un bandwidth de 112kbps, una prioridad menor que la que determina el tráfico de VoIP y va a identificar este tráfico por el puerto 6666.

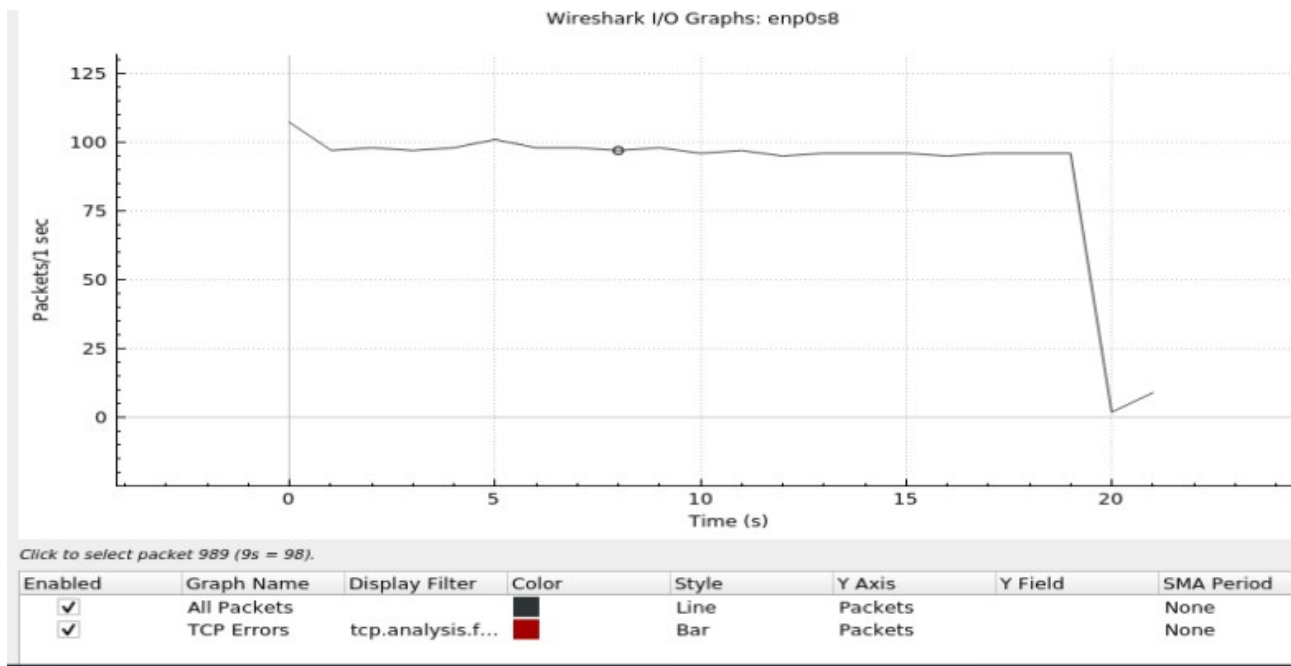
Para probarlo, generar el siguiente patrón de tráfico, y monitorice el tráfico que se recibe:

- Genere un flujo de VoIP.

```

root@manureina-virtualbox:/home/manureina# ITGSend -a 172.17.1.21 -rp 6667 -t 20000 VoIP
ITGSend version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Voice Codec: G.711.1
Framesize: 80.00
Samples: 1
Packets per sec.: 100
VAD: No
Started sending packets of flow ID: 1
Finished sending packets of flow ID: 1

```

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
16	0.050667126	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92
17	0.060701741	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92
18	0.071248104	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92
19	0.081224388	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92
20	0.091391746	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92
21	0.101800966	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92
22	0.112287264	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92
23	0.122421716	10.7.1.51	172.17.1.21	UDP	134	56009 → 6667 Len=92

Frame 19: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface enp0s8, id 0
 Ethernet II, Src: PcsCompu_8a:8a:27 (08:00:27:8a:8a:27), Dst: PcsCompu_c2:d6:f6 (08:00:27:c2:d6:f6)
 Internet Protocol Version 4, Src: 10.7.1.51, Dst: 172.17.1.21
 User Datagram Protocol, Src Port: 56009, Dst Port: 6667
 Destination Port: 6667
 Length: 100
 Checksum: 0xe01b [unverified]
 [Checksum Status: Unverified]
 [Stream index: 0]

```

0000  08 00 27 c2 d6 f6 08 00 27 8a 8a 27 08 00 45 00  ..x..@..?..p...3..
0010  00 78 b4 a4 40 00 3f 11 ce 70 0a 07 01 33 ac 11  ..x..@..?..p...3..
0020  01 15 da c9 1a 0b 00 64 e0 1b 00 00 00 01 00 00  ..x..@..?..p...3..
0030  00 08 00 00 34 d0 00 03 a5 ae 90 b5 38 be 56 66  ..x..@..?..p...3..
0040  f2 d7 48 b4 e7 a8 25 8f b5 98 48 5e 64 c3 86 d8  ..x..@..?..p...3..
0050  30 f4 2b 64 9c 0b e2 dc d0 73 92 0a 33 e8 ef a5  ..x..@..?..p...3..
0060  40 38 5a 29 e0 7f 38 17 97 01 75 fc c4 fb d5 74  ..x..@..?..p...3..
0070  71 80 d8 0e 8c bb ea 5d af fd e6 e2 e6 d6 88 28  ..x..@..?..p...3..
0080  10 e2 d0 70 e1 88                                     ..x..@..?..p...3..
  
```

- Genere 3 flujos diferentes de tráfico UDP a distintos puertos, con una tasa de 800 kbps cada uno.

```

-a 172.17.1.21 -rp 6666 -t 20000 -C 1000 -c 819,2 -T UDP Quake3
-a 172.17.1.21 -rp 6665 -t 20000 -C 1000 -c 819,2 -T UDP Quake3
-a 172.17.1.21 -rp 6664 -t 20000 -C 1000 -c 819,2 -T UDP Quake3
  
```

```

root@manureina-virtualbox:/home/manureina# ITGSend script.txt -l script.log
ITGSend version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Started sending packets of flow ID: 3
Started sending packets of flow ID: 1
Started sending packets of flow ID: 2
Finished sending packets of flow ID: 3

Finished sending packets of flow ID: 2
Finished sending packets of flow ID: 1

```

No.	Time	Source	Destination	Protocol	Length	Info
208	0.459141788	10.7.1.51	172.17.1.21	UDP	114	43685 → 6666 Len=72
209	0.459141835	10.7.1.51	172.17.1.21	UDP	106	43685 → 6666 Len=64
210	0.462973253	10.7.1.51	172.17.1.21	UDP	103	49470 → 6665 Len=61
211	0.464633479	10.7.1.51	172.17.1.21	UDP	103	51202 → 6664 Len=61
212	0.466307665	10.7.1.51	172.17.1.21	UDP	107	49470 → 6665 Len=65
213	0.466308041	10.7.1.51	172.17.1.21	UDP	107	51202 → 6664 Len=65
214	0.470383238	10.7.1.51	172.17.1.21	UDP	104	43685 → 6666 Len=62
215	0.477541808	10.7.1.51	172.17.1.21	UDP	103	49470 → 6665 Len=61

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s8, id 0

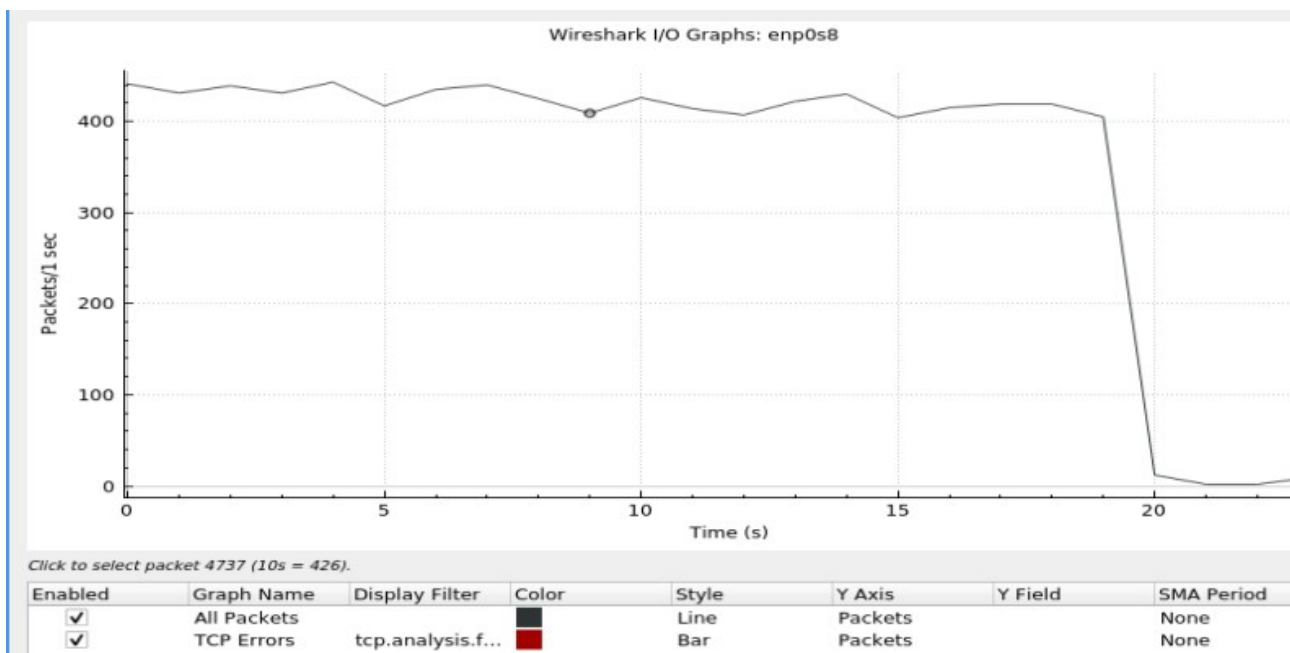
Ethernet II, Src: PcsCompu_8a:8a:27 (08:00:27:8a:8a:27), Dst: PcsCompu_c2:d6:f6 (08:00:27:c2:d6:f6)

Internet Protocol Version 4, Src: 10.7.1.51, Dst: 172.17.1.21

Transmission Control Protocol, Src Port: 43356, Dst Port: 9000, Seq: 0, Len: 0

Source Port: 43356
Destination Port: 9000
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)

Offset	Hex	ASCII
0000	08 00 27 c2 d6 f6 08 00 27 8a 8a 27 08 00 45 00E.
0010	00 3c 88 b1 40 00 3f 06 fa aa 0a 07 01 33 ac 11	<..@-?....3..
0020	01 15 a9 5c 23 28 da 34 da ee 00 00 00 00 a0 02	..#\(-4.....
0030	fa f0 65 7e 00 00 02 04 05 b4 04 02 08 0a 03 87	..e~.....
0040	aa 01 00 00 00 00 01 03 03 07



```

Flow number: 2
From 10.7.1.51:49470
To 172.17.1.21:6665
-----
Total time = 19.999715 s
Total packets = 2761
Minimum delay = 0.000000 s
Maximum delay = 0.000000 s
Average delay = 0.000000 s
Average jitter = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received = 175850
Average bitrate = 70.341002 Kbit/s
Average packet rate = 138.051967 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0.000000 pkt
-----

***** TOTAL RESULTS *****
-----
Number of flows = 3
Total time = 20.005863 s
Total packets = 8461
Minimum delay = 0.000000 s
Maximum delay = 0.000000 s
Average delay = 0.000000 s
Average jitter = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received = 538795
Average bitrate = 215.454839 Kbit/s
Average packet rate = 422.926019 pkt/s
Packets dropped = 0 (0.00 %)

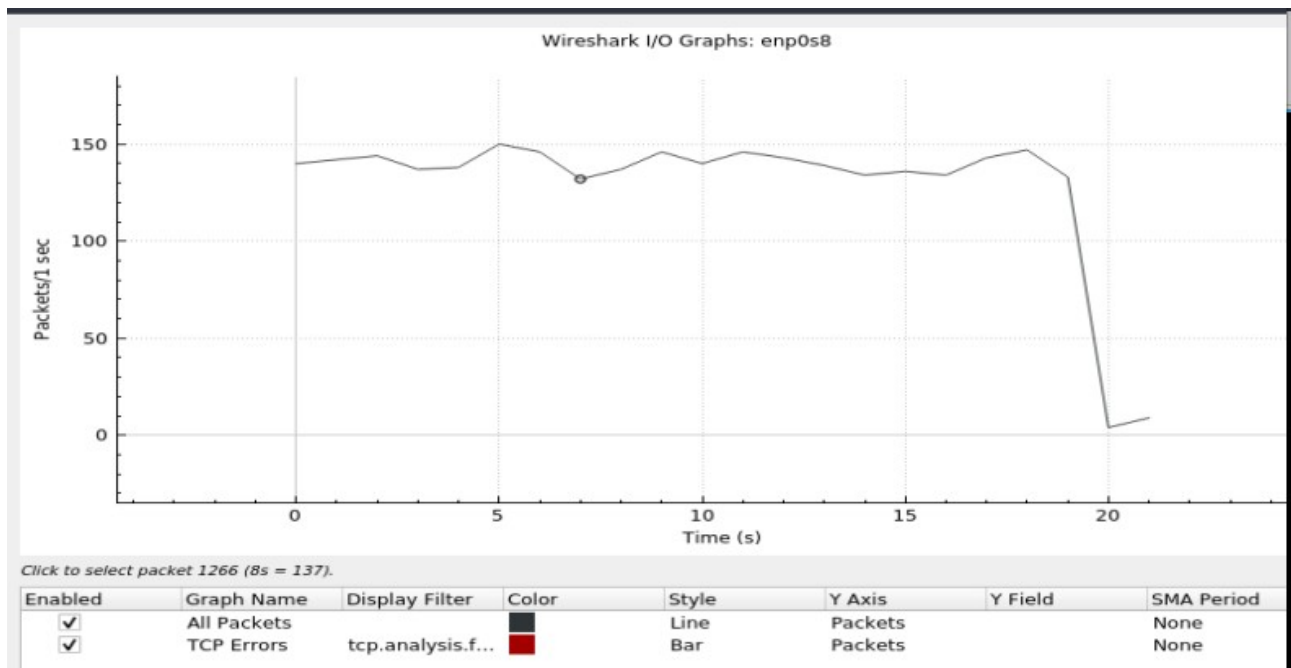
```

- Genere un flujo de tráfico Quake3.

```

-a 172.17.1.21 -rp 6666 -t 20000 Quake3

```



76	0.500916972	10.7.1.51	172.17.1.21	UDP	106	44074 → 6666	Len=64
77	0.500917834	10.7.1.51	172.17.1.21	UDP	102	44074 → 6666	Len=60
78	0.501779228	10.7.1.51	172.17.1.21	UDP	103	44074 → 6666	Len=61
79	0.502476085	10.7.1.51	172.17.1.21	UDP	108	44074 → 6666	Len=66
80	0.506083245	10.7.1.51	172.17.1.21	UDP	104	44074 → 6666	Len=62
81	0.517383072	10.7.1.51	172.17.1.21	UDP	106	44074 → 6666	Len=64
82	0.524544799	10.7.1.51	172.17.1.21	UDP	105	44074 → 6666	Len=63
83	0.535103400	10.7.1.51	172.17.1.21	UDP	104	44074 → 6666	Len=62

▶ Frame 80: 104 bytes on wire (832 bits), 104 bytes captured (832 bits) on interface enp0s8, id 0
 ▶ Ethernet II, Src: PcsCompu_8a:8a:27 (08:00:27:8a:8a:27), Dst: PcsCompu_c2:d6:f6 (08:00:27:c2:d6:f6)
 ▶ Internet Protocol Version 4, Src: 10.7.1.51, Dst: 172.17.1.21
 ▶ User Datagram Protocol, Src Port: 44074, Dst Port: 6666
 Source Port: 44074
 Destination Port: 6666
 Length: 70
 Checksum: 0x69ee [unverified]
 [Checksum Status: Unverified]
 [Stream index: 0]

0000	08 00 27 c2 d6 f6 08 00	27 8a 8a 27 08 00 45 00	..ZC@?@...E
0010	00 5a 43 13 40 00 3f 11	40 20 0a 07 01 33 ac 11	..@.??@...3..
0020	01 15 ac 2a 1a 0a 00 46	69 ee 00 00 00 01 00 00	..*..F1.....
0030	00 45 00 00 3d 45 00 0e	73 cb db 5a 0f 78 43 da	..E==E...s..Z..xC
0040	66 ed ae c9 f9 ca 66 28	6d e4 28 01 9f c3 54 d0	f....f(m...(T
0050	b9 6a 76 b0 97 b9 75 08	e9 d0 62 f8 c9 a5 d3 30	..jv...u...b....0
0060	13 82 79 0e 4d 60 36 3b		..y-M'6;

```

Flow number: 1
From 10.7.1.51:44074
To 172.17.1.21:6666
-----
Total time = 19.997950 s
Total packets = 2794
Minimum delay = 0.000000 s
Maximum delay = 0.000000 s
Average delay = 0.000000 s
Average jitter = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received = 177478
Average bitrate = 70.998477 Kbit/s
Average packet rate = 139.714321 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0.000000 pkt
-----

***** TOTAL RESULTS *****
-----
Number of flows = 1
Total time = 19.997950 s
Total packets = 2794
Minimum delay = 0.000000 s
Maximum delay = 0.000000 s
Average delay = 0.000000 s
Average jitter = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received = 177478
Average bitrate = 70.998477 Kbit/s
Average packet rate = 139.714321 pkt/s
Packets dropped = 0 (0.00 %)

```